

One page statement describing the overall plan and description of your customized game.

### Project Phase 1: Escape SFU

Our game revolves around our main character, an SFU Student, stuck on campus after hours and trying to escape without being caught by their professor, their TAs (possibly the Dean? More types and instantiations of enemies, the better), and campus security looking to catch them in the act. The SFU student navigates through campus, attempting to collect all the regular bonuses like textbooks and lecture notes. The bonus rewards, which in this case are bonus marks and exam solutions, appear sporadically and are only available for limited amounts of time, but are not needed to complete the game. The student maneuvers through inanimate punishments such as mark deductions, security cameras, and pop quizzes while avoiding the aforementioned enemies. The student's ultimate goal is to make it to the last cell of the board (the campus' exit) with all textbooks and lecture notes in hand.

## **Part 1:**

Game: SFU Break

1. Main Character: SFU Student

2. Enemies

-Moving Enemy: Professor, TAs, Dean, Campus Security

-Punishments (Static Enemy): Security Camera, Mark Deductions

3. Rewards:

- Normal: Textbooks and Lecture Notes

-Bonus: Bonus Marks and Exam Solutions

4. Barriers: Cells and Doors

5. Board: SFU Campus Layout (Maze Like)

Overview: Escape the Campus with all Textbooks and Lecture Notes in Hand

## **Part 2:**

**Use cases: Setting**

Primary actor: User

The goal in context: Set the game mode from easy to hard or audio volume depending on the user's wants.

Preconditions: It only can be accessed through the start menu and can not be changed during the game.

Trigger: When the user selects the game mode and saves it, the speed of the enemies changes depending on the mode.

Scenario:

1. User: observes setting
2. User: Sets the game mode in the options easy, normal, or hard
3. User: click the save button and exit the setting

Exceptions:

1. The primary setting is in normal mode
2. When the user does not click the save button, does not change the mode
3. The primary volume setting is medium sound.

Priority:

When available: In the main menu

Frequency of use: As many times the user wants

Channel to the actor: The setting interface

Secondary actors: Support technician, audio system

Channels to secondary actors:

Support technician: Phone line, email

Audio system: audio interfaces

Open issues:

1. Should there be a way to change the volume settings while the game is in progress?

### **Use case: The player starts a new game**

Primary actor: Player

The goal in context: To create a new game

Preconditions: The player is logged in and on the game's main menu

Trigger: The player clicks on the "Play" button on the main menu

Scenario:

1. User: The player clicks the "Play" button
2. System: The game generates a new game and displays the main character, the game grid, enemies and rewards

Exceptions:

1. Game world generation fails: the game displays an error message and prompts the player to try again.

Priority: High, must be implemented before the player can play the game

When available: At all times

Frequency of use: Once per player per game

Channel to the actor: Game interface

Secondary actors: None

Channels to secondary actors: None

Open issues:

1. Should the game instantly start, or should there be a countdown?

### **Use cases: move character**

Primary actor: The user

The goal in context: To move the main character in the direction of the inputted corresponding arrow keys and progress through the objectives.

Precondition: The game is currently in progress, and the player character can move

Trigger: The player uses the arrow keys to move the main character through the maze-like game board while avoiding the moving enemies and barrier

Scenario:

1. User: Use the arrow keys to move the character

Exceptions:

1. Arrow keys do not work: may need to restart the game.
2. Invalid key pressed: main character will not move.
3. Move is invalid (move to a cell that is blocked by a wall or a barrier): main character will not move.
4. The game has ended: The main character can no longer move.

Priority: High, must be implemented when the player can play the game

When available: While the game is in progress

Frequency of use: Unlimited while game is in progress

Channel to the actor: Game interface

Secondary actors: None

Channels to secondary actors: None

Open issues:

1. Should there be a visual indication if the user attempts an invalid move?

### **Use cases: Collect Regular Reward**

Primary actor: User

The goal in context: Allows the player character to collect a regular reward

Preconditions: The player character moves to a cell containing a regular reward.

Trigger: The player score is updated with the reward value, and the new score is displayed on the screen.

Scenario:

1. User: controls the character
2. User: find the rewards and collects it
3. System: The reward is removed from the cell, and the player score is updated

Exceptions:

1. Quest giver is not available: the game prompts the player to try again later
2. The game server is down: the game displays an error message and encourages the player to try again later.

Priority: High, must be implemented during the game

When available: During active quests

Frequency of use: regular keep updates after the user collects the reward

Channel to the actor: The setting interface

Secondary actors: Support technician, audio system

Channel to the actor: Game interface

Secondary actors: None

Channels to secondary actors: None

Open issues:

1. Should there be an additional visual indicator to show that the user has collected a reward.

### **Use cases: Collect Bonus Reward**

Primary actor: User

The goal in context: Allows the player character to collect a bonus reward

Preconditions: A bonus reward appears randomly on the board.

Trigger: The player score is updated with the bonus value, and the new score is displayed on the screen.

Scenario:

1. User: controls the character
2. User: find the bonus rewards and collects it
3. System: The reward is removed from the cell, and the player score is updated

Exceptions:

1. Quest giver is not available: the game prompts the player to try again later.
2. The game server is down: the game displays an error message and encourages the player to try again later.

Priority: High, must be implemented during the game

When available: During active quests

Frequency of use: Unique

Channel to the actor: The setting interface

Secondary actors: Support technician, audio system

Channel to the actor: Game interface

Secondary actors: None

Channels to secondary actors: None

Open issues:

1. Should there be an additional visual indicator to show that the user has collected a bonus reward.

### **Use case: The player completes a quest**

Primary actor: User

The goal in context: Handles the case where the player successfully reaches the end cell after collecting all regular rewards.

Preconditions: The player character is at the end cell and has collected all regular rewards.

Trigger: The player meets the quest requirements

Scenario:

1. User: collected all regular rewards.
2. System: The game verifies that the quest objectives have been met
3. System: The player wins the game, and the final score and time are displayed on the screen.

Exceptions:

1. The player does not complete all quest objectives: the game prompts the player to finish the quest.
2. Quest giver is unavailable: the game prompts the player to try again later.
3. The game server is down: the game displays an error message and prompts the player to try again later.

Priority: High quest completion is a core game mechanic.

When available: During active quests

Frequency of use: Varies by player and quest

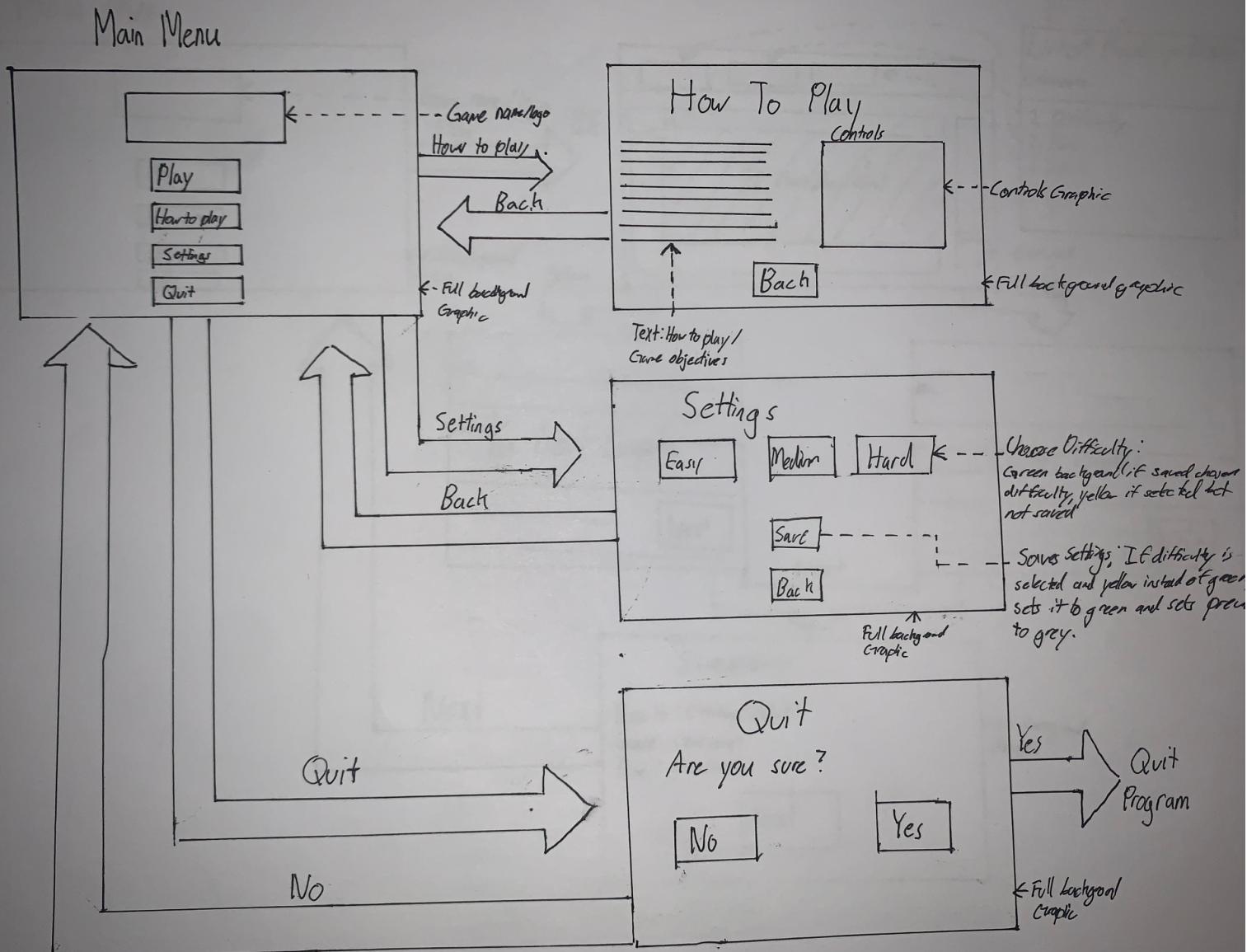
Channel to the actor: Game interface

Secondary actors: Quest giver, game server

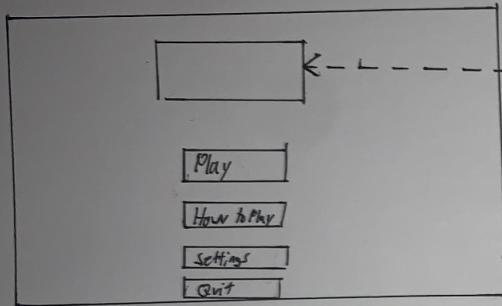
Channels to secondary actors: In-game messages, game server connection  
Open issues: None.

### Part 3:

#### UI Mockups:



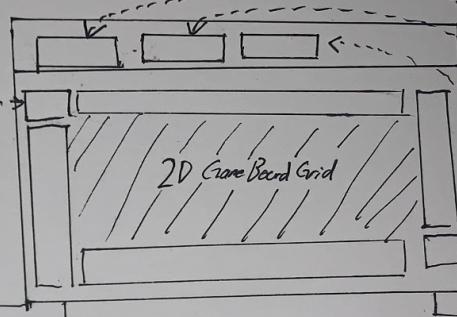
## Main Menu



Game name/logo  
Play

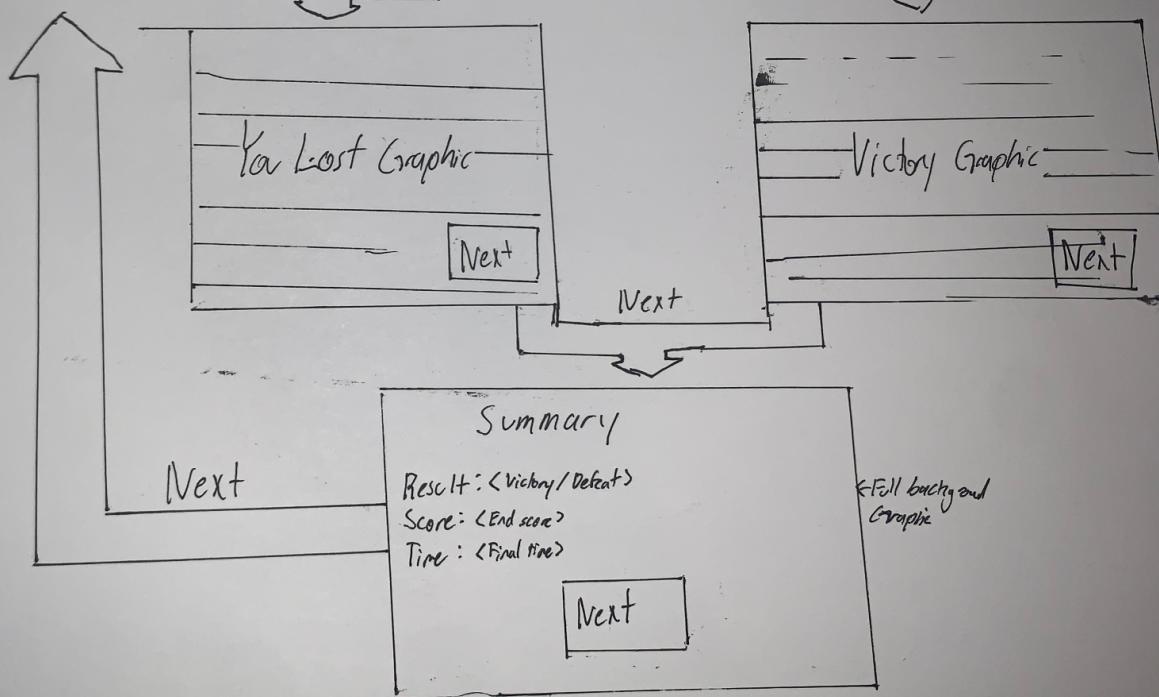
< Full/Background Graphic

Defeat



### List of Heads-up Display Elements

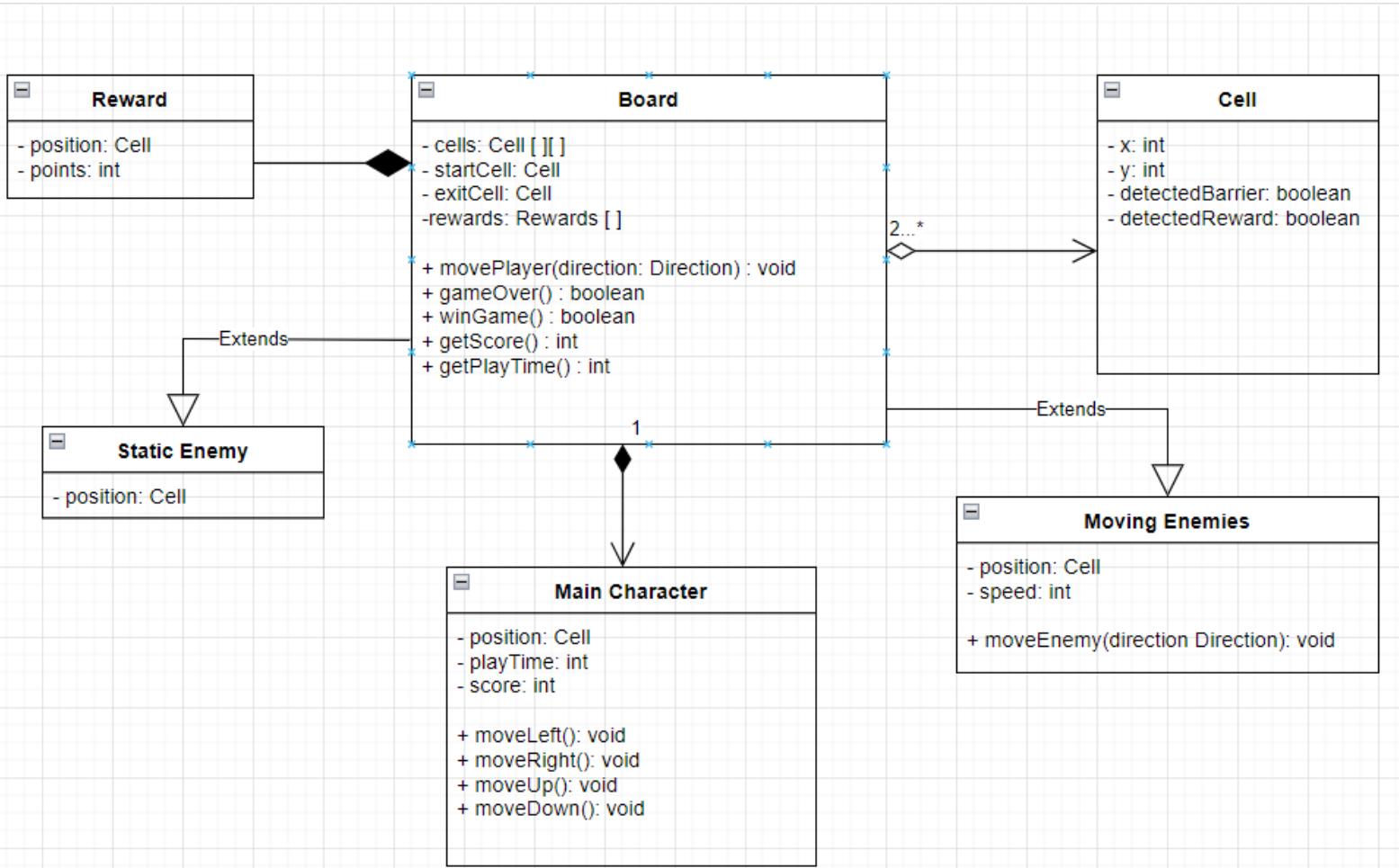
1. Difficulty
2. Time
3. Score



## Part 4: UML Diagrams

[https://drive.google.com/file/d/1ixHOhpuzIV\\_Y8TOuorqxqP0cv18OW-N/view?usp=sharing](https://drive.google.com/file/d/1ixHOhpuzIV_Y8TOuorqxqP0cv18OW-N/view?usp=sharing)

Current (Latest) UML:



Do we need Game Logic? That takes User Input from Keyboard and manage States of the Game.

It is just a rough sketch of our current planning.

The UML Diagram above will be subject to changes when we implement our logic and games.

## **Brainstorming:**

(Some Ideas) We can change this part anytime

### Game: Temple Run

1. Main Character: Treasure Hunter
2. Enemies
  - Moving Enemy: Bats, Spiders, Thieves
  - Punishments: Hidden Traps, Mines
3. Rewards:
  - Normal: Gold, Rare Artifacts
  - Bonus: Buff, Speed Increase, Defense
4. Barriers: Stone Walls, Dark Areas, Water or Lava
5. Board: Ancient Temple

Overview: Steal as much Golds and escape without getting killed.

### Game: Prison (SFU) Break

1. Main Character: Inmate
  2. Enemies
    - Moving Enemy: Prison Guards
    - Punishments (Static Enemy): Security Camera
  3. Rewards:
    - Normal: Keys, Tools, Money
    - Bonus: Speed Increase, ...
  4. Barriers: Cells and Doors
  5. Board: Prison with Cells, Corridors, Outdoor Areas
- Overview: Escape the Prison without being caught by the Guards/Camera.